Chapter 4
*Autoregressive and Moving-Average Models II*

### 4.1 Box-Jenkins Model Selection

The Box-Jenkins approach to modeling involves three iterative steps to identify the appropriate ARIMA process to estimate and forecast a univariate time series.

1. **Model Selection** or identification stage involves visualizing the time plot of the series, the autocorrelation and the partial correlation function. Plotting the time path of \( \{y_t\} \) provides useful information about outliers, missing values, and structural breaks in the data generating process. Nonstationary variables may have a pronounced trend or appear to meander without a constant long-run mean or variance.

2. **Parameter Estimation** involves finding the values of the model coefficients which provide the best fit to the data. In this stage the goal is to select a stationary and parsimonious model that has a good fit.

3. **Model Checking** involves ensuring that the residuals from the estimated model mimic a white-noise process.

### 4.2 The Autocorrelation Function

We mentioned in the previous chapter that for a process to be covariance stationary, we need that the mean is stable over time

\[
E(y_t) = E(y_{t-s}) = \mu \quad \forall t, s
\]  

(4.1)

its variance is stable over time

\[
E[(y_t - \mu)^2] = E[(y_{t-s} - \mu)^2] = \sigma^2 \quad \forall t, s
\]  

(4.2)

and that its autocovariance function also does not depend on time
\[ E[(y_t - \mu)(y_{t-s} - \mu)] = E[(y_{t-j} - \mu)(y_{t-j-s} - \mu)] = \gamma_s \quad \forall t, s, j \quad (4.3) \]

This last equation indicates that the autocovariance function \( \gamma_s \) depends only on the displacement \( s \) and not on the time \( t \). The autocovariance function is important because it provides a basic summary of cyclical dynamics in a covariance stationary series. Note that the autocovariance function is symmetric, \( \gamma_s = \gamma_{-s} \). Also note that \( \gamma_0 = E[(y_t - \mu)(y_{t-0} - \mu)] = E[(y_t - \mu)^2] = \sigma^2 \).

For the same reasons one normally uses the correlation coefficient rather than the variance, it is useful to define the autocorrelation function as follows

\[ \rho_s = \frac{\gamma_s}{\gamma_0} \quad (4.4) \]

which is the same we defined before in Equation 3.5. The formula for the autocorrelation is just the usual correlation formula, tailored to the correlation between \( y_t \) and \( y_{t-s} \). The sample analog is,

\[ \hat{\rho}_s = \frac{\sum_{t=s+1}^{T} (y_t - \bar{y})(y_{t-s} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2} \quad (4.5) \]

this estimator is the sample autocorrelation function or correlogram.

It is of interest to know whether a series is a reasonable approximation to a white-noise, which is to say that all its autocorrelations are 0 in the population. If a series is white-noise, then the distribution of the sample autocorrelations in large samples is

\[ \hat{\rho}_s \sim N\left(0, \frac{1}{T}\right) \quad (4.6) \]

That is, they are approximately normally distributed. In practice, 95% of the sample autocorrelations should fall in the interval \( \pm \frac{2}{\sqrt{T}} \). We can rewrite Equation 4.6 as

\[ \sqrt{T}\hat{\rho}_s \sim N(0, 1). \quad (4.7) \]

Then, squaring both sides we obtain

\[ T\hat{\rho}_s^2 \sim \chi^2_1. \quad (4.8) \]

It can be shown that the sample autocorrelations at various displacements are approximately independent of one another. Hence, we can obtain the \textit{Box-Pierce Q-statistic} as the sum of \( m \) independent \( \chi^2 \) variables,

\[ Q_{BP} = T \sum_{s=1}^{m} \hat{\rho}_s^2 \quad (4.9) \]

which follows a \( \chi^2_m \) distribution under the null that \( y \) is white noise. As slight modification, designed to follow more closely the \( \chi^2 \) distribution in small samples is the \textit{Ljung-Box Q-statistic},...
4.3 The Partial Autocorrelation Function

\[ Q_{LB} = T(T + 2) \sum_{s=1}^{m} \left( \frac{1}{T-s} \right) \hat{\rho}_s^2. \] (4.10)

Under the null that \( y \) is white noise, \( Q_{LB} \) is approximately distributed as a \( \chi_m^2 \) random variable.

4.3 The Partial Autocorrelation Function

The partial autocorrelation function, \( p(s) \), is just the coefficient of \( y_{t-s} \) in a population linear regression of \( y_t \) on \( y_{t-1}, y_{t-2}, \ldots, y_{t-s} \). That is,

\[ \hat{y}_t = \hat{c} + \hat{\beta}_1 y_{t-1} + \cdots + \hat{\beta}_s y_{t-s}, \] (4.11)

then the sample partial correlation at displacement \( s \) is

\[ \hat{p}_s = \hat{\beta}_s. \] (4.12)

If the series is white noise, approximately 95% of the sample partial autocorrelations should fall in the interval \( \pm \frac{2}{\sqrt{T}} \).

Figure 4.1 shows the autocorrelation and the partial autocorrelation functions of a white-noise process presented in Figure 3.1. This one was obtained using:

```stata
ac white, saving(acwhite)
pac white, saving(pacwhite)
gr combine acwhite.gph pacwhite.gph, col(1) ///
iscale(0.7) fysize(100) ///
title("ACF and PACF for a White-Noise Process")
```

Because it is white-noise, by definition it is uncorrelated over time. Hence, all population autocovariances and autocorrelations should be zero beyond displacement 0. The figure shows that most of the sample autocovariances and autocorrelations fall within the 95% confidence bands.\(^1\)

To obtain the correlogram that comes with the Ljung-Box Q-statistic (Portman-teau (Q)) we need to type

```stata
corrgram white, lags(20)
```

to obtain

<table>
<thead>
<tr>
<th>LAG</th>
<th>AC</th>
<th>PAC</th>
<th>Q</th>
<th>Prob&gt;Q [Autocorrelation]</th>
<th>Partial Autocor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0048</td>
<td>-0.0047</td>
<td>0.00346</td>
<td>0.9531</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-0.2227</td>
<td>-0.2230</td>
<td>7.6432</td>
<td>0.0219</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-0.0019</td>
<td>-0.0040</td>
<td>7.6437</td>
<td>0.0540</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>0.1439</td>
<td>0.1003</td>
<td>10.878</td>
<td>0.0280</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0.0383</td>
<td>-0.0405</td>
<td>11.086</td>
<td>0.0497</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^1\) For the variance of \( \hat{\rho}_s \) given by Bartlett's formula for MA(\( q \)) processes used in Stata, see the Stata Manual. Moreover, Stata also has the Yule-Walker option to estimate the partial autocorrelations.
Nearly all Q-statistics fail to reject the null of a white-noise process.

### 4.3.1 Examples using Simulated Processes

Now we present the autocorrelation and the partial autocorrelation functions of the MA(1) and AR(1) simulated processes in Equations 3.7 and 3.15. Just few observations. (1) The $y_1$ simulated process have very short-lived dynamics and cannot be distinguished from a white-noise process. (2) Most of the action in the MA(1) processes appear in the partial autocorrelations. When $\theta$ is relatively large and pos-
4.4 Model Selection Criteria

As is well known in multiple regression analysis adding regressors to the equation always improves the model fit, as measured by the $R^2$. Hence, a model with additional lags for $p$ and/or $q$ will certainly fit better the current sample. This will not make the model better, for example, for forecasting. In addition, more lags mean less degrees of freedom. The most commonly used model selection criteria to obtain a parsimonious model are the Akaike Information Criterion (AIC) and the Schwartz Bayesian Criterion (SBC). Their formulas are given by
4.5 Parameter Instability and Structural Change

One key assumption in nearly all time-series models is that the underlying structure of the data-generating process does not change. That is, the parameters we are estimating are constant over time and do not have subscript \( t \) (e.g., \( \theta_t, \beta_t \)). However, in some cases, there may be reasons to believe that there is a structural break in the data-generating process. For example, when estimating a model of airline demand we may suspect that the estimated parameters may be different after and before the tragedy of 9/11.
4.5.1 Testing for Structural Change

It is easy to test for the existence of a structural break using a Chow test. The idea is to estimate the same ARMA model after and before the suspected break, then if the two models are not significantly different we conclude that there is no structural change in the data-generating process.

Let $SSR_1$ be the sum of squared residuals of the model prior the suspected break, $SSR_2$ be the sum of squared residuals of the model fitted after the suspected break, and $SSR$ using the whole sample. The Chow test uses the following $F$-statistic

$$F_{n,T-2n} = \frac{(SSR - SSR_1 - SSR_2)/n}{(SSR_1 + SSR_2)/(T-2n)}$$

where $n$ is the number of estimated parameters and $T$ is the sample size. The larger the $F$ the more restrictive the assumption that the coefficients are equal.

4.5.2 Endogenous Breaks

The Chow test assumes the date of the brake is given exogenously. When the date of the break is not pre-specified and it is rather determined by the data, then we
call it an endogenous break. A simple way to let the data select the break is to run multiple Chow test at different break points and then select the one with the largest $F$-statistic. While this process works intuitively, the search of the most likely break means that the $F$-statistic for the null hypothesis of no break is inflated. This means that the distribution of this $F$-statistic is non-standard. To solve this Hansen (1997) explain how to use bootstrapping methods to obtain the critical values.

4.5.3 Parameter Instability

Parameter instability is often assessed using recursive estimation procedures that allow tracking how parameters change over time. The idea in recursive parameter estimation is that we estimate the model many times. Consider, for example, the following model

$$y_t = \sum_{i=1}^{k} \beta_i x_{i,t} + \epsilon_t$$  \hspace{1cm} (4.16)

for $t = 1, 2, \ldots, T$. Instead of estimating the model using the whole data set, we estimate it via OLS with the first $k$ observations, then again with the first $k+1$ observations, and so on until the sample is exhausted. At the end we will have recursive parameter estimates $\hat{\beta}_{i,t}$ for $t = k, k+1, \ldots, T$ and $i = 1, 2, \ldots, k$. 

Fig. 4.5 ACF and PACF for the Simulated Process $z_{1,t} = +0.9 \cdot z_{1,t-1} + \epsilon_t$
At each \( t, t = k, k + 1, \ldots, T - 1 \), we can compute the 1-step-ahead forecast, 
\[
\hat{y}_{t+1} = \sum_{i=1}^{k} \hat{\beta}_i x_{t+i+1}.
\]
The corresponding forecast errors, or recursive residuals, are 
\[
\hat{e}_{t+1} = y_{t+1} - \hat{y}_{t+1}.
\]
We then standardize the recursive residuals to take into account the fact that the variance of the 1-step-ahead forecast errors changes over time as the sample size grows. Let \( w_{t+1} \) denote the standardized recursive residuals. Then, the cumulative sum “CUSUM" of the standardized recursive residuals is particularly useful in assessing parameter stability.

\[
CUM_{st} = \sum_{s=k}^{t} w_{s+1}, \quad t = k, k + 1, \ldots, T - 1
\] (4.17)

The CUSUM is just a sum of i.i.d. \( N(0,1) \) random variables. Probability bounds for the CUSUM have been tabulated, and we can examine the time-series plots of the CUSUM and its 95% probability bounds, which grow linearly and are centered at 0. If the CUSUM violates the bounds at any point, there is evidence of parameter instability. The Stata commands to implement this test and obtain Figure 4.7 are as follows.

```stata
ssc install cusum6
clear
tsset t
generate ibmadj = ibm - irx
generate spxadj = spx - irx
cusum6 ibmadj spxadj, cs(cusum) lw(lower) uw(upper)
twoway line cusum lower upper t, scheme(sj) ///
title("CUSUM")
```
Figure 4.7 shows evidence of parameter instability around $t = 102$. In these data we have daily returns to IBM stock (ibm), the S&P 500 (spx), and short-term interest rates (irx), and we are estimating the beta for IBM.

Using the same data we can see search for specific coefficient’s stability using the `rolling` command in Stata. We want to create a series containing the beta of IBM by using the previous 200 trading days at each date. We will also record the standard errors, so that we can obtain 95% confidence intervals for the betas. The detailed Stata command to obtain Figure 4.8 is

```stata
clear
tset t
generate ibmadj = ibm - irx
generate spxadj = spx - irx
rolling _b _se, window(200) saving(betas, replace) keep(date): regress ibmadj spxadj
use betas, clear
label variable _b_spxadj "IBM beta"
generate lower = _b_spxadj - 1.96*_se_spxadj
generate upper = _b_spxadj + 1.96*_se_spxadj
twoway (line _b_spxadj date) (line lower upper date) if date>=td("1oct2003"), scheme(sj) ytitle("Beta") title("Rolling Regression to Reveal Parameter Instability")
```

---

**Fig. 4.7 CUSUM Test**

---

Figure 4.7 shows evidence of parameter instability around $t = 102$. In these data we have daily returns to IBM stock (ibm), the S&P 500 (spx), and short-term interest rates (irx), and we are estimating the beta for IBM.

Using the same data we can see search for specific coefficient’s stability using the `rolling` command in Stata. We want to create a series containing the beta of IBM by using the previous 200 trading days at each date. We will also record the standard errors, so that we can obtain 95% confidence intervals for the betas. The detailed Stata command to obtain Figure 4.8 is

```stata
clear
tset t
generate ibmadj = ibm - irx
generate spxadj = spx - irx
rolling _b _se, window(200) saving(betas, replace) keep(date): regress ibmadj spxadj
use betas, clear
label variable _b_spxadj "IBM beta"
generate lower = _b_spxadj - 1.96*_se_spxadj
generate upper = _b_spxadj + 1.96*_se_spxadj
twoway (line _b_spxadj date) (line lower upper date) if date>=td("1oct2003"), scheme(sj) ytitle("Beta") title("Rolling Regression to Reveal Parameter Instability")
```
4.6 Forecasts

Maybe the most popular use of ARMA models is to forecast future values of the \( \{y_t\} \) series. Assume that the actual data-generating process and the current and part realizations of the \( \{\varepsilon_t\} \) and \( \{y_t\} \) sequences are known. Consider the following AR(1) model:

\[
y_t = a_0 + a_1 y_{t-1} + \varepsilon_t \tag{4.18}
\]

which can be written as

\[
y_{t+1} = a_0 + a_1 y_t + \varepsilon_{t+1} \tag{4.19}
\]

Knowing the data-generating process means knowing \( a_0 \) and \( a_1 \). Then we can forecast \( \{y_{t+1}\} \) conditioning on the information set available at time \( t \),

\[
E_t(y_{t+1}) = a_0 + a_1 y_t \tag{4.20}
\]

where we use \( E_t(y_{t+j}) \) to denote the conditional expectation of \( y_{t+j} \) given the information set available at time \( t \). That is, \( E_t(y_{t+j}) = E(y_{t+j}|y_t, y_{t-1}, y_{t-2}, \ldots, \varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots) \). The forecast two periods ahead is:

\[
E_t(y_{t+2}) = a_0 + a_1 E_t(y_{t+1}) \tag{4.21}
\]

\[
E_t(y_{t+2}) = a_0 + a_1(a_0 + a_1 y_t)
\]
This is the two-step ahead forecast. With additional iterations, forecasts can be
constructed to obtain \( y_{t+j} \), given that \( y_{t+j-1} \) is already forecasted,

\[
E_t(y_{t+j}) = a_0 + a_1 E_t(y_{t+j-1}),
\]

\[
E_t(y_{t,j}) = a_0 (1 + a_1 + a_1^2 + \cdots + a_1^{j-1}) + a_1^j y_t.
\]

Equation 4.23 is the forecast function, which shows all the \( j \)-step-ahead forecasts as
a function of the information set at time \( t \). Since \(|a_1| < 1\),

\[
\lim_{j \to \infty} E_t(y_{t+j}) = \frac{a_0}{1 - a_1}
\]

This last equation shows that for any stationary ARMA model, the conditional fore-
cast of \( y_{t+j} \) converges to the unconditional mean as \( j \to \infty \).

We define the forecast error \( e_t(j) \) for the \( j \)-step-ahead forecast as

\[
e_t(j) \equiv y_{t+j} - E_t(y_{t+j}).
\]

In the one-step-ahead forecast, the error is \( e_t(1) = y_{t+1} - E_t(y_{t+1}) = \varepsilon_{t+1} \), which
by definition is unforecastable portion of \( y_{t+1} \) given the information set at \( t \). The
two-step-ahead forecast error is \( e_t(2) = a_1 (y_{t+1} - E_t(y_{t+1})) + \varepsilon_{t+2} = \varepsilon_{t+2} + a_1 \varepsilon_{t+1} \),
while the \( j \)-step-ahead forecast error is

\[
e_t(j) = \varepsilon_{t+j} + a_1 \varepsilon_{t+j-1} + a_1^2 \varepsilon_{t+j-2} + a_1^3 \varepsilon_{t+j-3} + \cdots + a_1^{j-1} \varepsilon_{t+1}.
\]

Notice that the mean of Equation 4.25 is zero, which means that the forecasts are
unbiased. If we look at the variance of the forecast error, we see that

\[
\text{var}[e_t(j)] = \sigma^2 \left[ 1 + a_1^2 + a_1^4 + a_1^6 + \cdots + a_1^{2(j-1)} \right].
\]

Note that the variance of the forecast error is increasing in \( j \). It is \( \sigma^2 \) for the one-
step-ahead forecast, \( \sigma^2 (1 + a_1^2) \) for the two-step-ahead forecast, and so on. When
\( j \to \infty \),

\[
\lim_{j \to \infty} \text{var}[e_t(j)] = \sigma^2 \frac{1}{1 - a_1}
\]

thus, the forecast error variance converges to the unconditional variance of the \( \{y_t\} \)
sequence.

If we assume that the sequence \( \{\varepsilon_t\} \) is normally distributed, we can place con-
dfidence intervals around the forecasts. For example, the 95% confidence interval for
the two-step-ahead forecast is

\[
a_0 (1 + a_1) + a_1^2 y_t \pm 1.96 \sigma (1 + a_1)^{1/2}.
\]
4.6 Forecasts

4.6.1 Forecast Evaluation

How do we evaluate forecasts? Consider the forecast of the value $y_{T+1}$ using Equation 4.19. Then the one-step-ahead forecast error is

$$e_T(1) = y_{T+1} - E_t(y_{T+1})$$  \hspace{1cm} (4.29)

$$= y_{T+1} - a_0 - a_1 y_T$$  \hspace{1cm} (4.30)

$$= e_{T+1}$$  \hspace{1cm} (4.31)

Since the forecast error $e_{T+1}$ is unforecastable, it appears that no other model can possibly provide a superior forecasting performance. The problem is that $a_0$ and $a_1$ need to be estimated from the sample. If we use the estimated model, the one-step-ahead forecast will be

$$E_t(y_{T+1}) = \hat{a}_0 + \hat{a}_1 y_T$$  \hspace{1cm} (4.32)

and the one-step-ahead forecast error will be

$$e_T(1) = y_{T+1} - (\hat{a}_0 + \hat{a}_1 y_T)$$  \hspace{1cm} (4.33)

which is clearly different from the error in Equation 4.31 because $a_0$ and $a_1$ are obtained imprecisely. Forecasts using overly parsimonious models with little parameter uncertainty can provide better forecasts than models consistent with the actual data-generating process.

One way to evaluate and compare forecasts is to use the mean square prediction error (MSPE). Suppose that we construct $H$ one-step-ahead forecasts from two different models. Let $f_{1i}$ be the forecast of model 1 and $f_{2i}$ be the forecast of model 2. Then the two series of forecast errors are $e_{1i}$ and $e_{2i}$. Then the MSPE of model 1 can be obtained with

$$\text{MSPE} = \frac{1}{H} \sum_{i=1}^{H} e_{1i}^2$$  \hspace{1cm} (4.34)

There are several methods to determine whether the MSPE from one model is statistically different than the MSPE from a second model. If we put the larger MSPE in the numerator, a standard recommendation is to use the F-statistic

$$F = \frac{\sum_{i=1}^{H} e_{1i}^2}{\sum_{i=1}^{H} e_{2i}^2}$$  \hspace{1cm} (4.35)

Large $F$ values imply that the forecast errors from the first model are larger than those of the second model. Under the null of equal forecasting performance, Equation 4.35 follows an $F$ distribution with $(H, H)$ degrees of freedom under these three assumptions:

1. Forecast errors follow a normal distribution with zero mean.
2. Forecast errors are not serially correlated.
3. Forecast errors are contemporaneously uncorrelated with each other.

A popular test for evaluating and comparing forecasts is the Diebold and Mariano (1995) testing procedure. The advantage of this test is that it relaxes assumptions 1 to 3 and allows for an objective function that is not quadratic. Let \( g(e_i) \) denote a loss function from a forecasting error in period \( i \). Notice that one typical case of this loss function would be the mean squared errors, where the loss is \( g(e_i) = e_i^2 \).

We can write the differential loss in period \( i \) from using model 1 versus model 2 as \( d_i = g(e_{1i}) - g(e_{2i}) \). Hence, the mean loss can be obtained with

\[
\bar{d} = \frac{1}{H} \sum_{i=1}^{H} [g(e_{1i}) - g(e_{2i})]
\]

Under the null of equal forecast accuracy, the value of \( \bar{d} \) is zero. Because \( \bar{d} \) is the mean of individual losses, under fairly weak conditions the central limit theorem implied that \( \bar{d} \) should follow a normal distribution. Hence, \( \bar{d} / \sqrt{\text{var}(\bar{d})} \) should follow a standard normal distribution. The problem in the implementation is that we need to estimate \( \text{var}(\bar{d}) \). Diebold and Mariano proceed in the following way. Let \( \gamma_i \) denote the \( i \)th autocovariance of the \( \bar{d} \) sequence, then, suppose that the first \( q \) values of \( \gamma_i \) are different from zero. The variance of \( \text{var}(\bar{d}) \) can be approximated using

\[
\text{var}(\bar{d}) = [\gamma_0 + 2\gamma_1 + \cdots + 2\gamma_q] (H-1)^{-1}
\]

Then the Diebold-Mariano (DM) statistic is given by

\[
DM = \frac{\bar{d}}{\sqrt{[\gamma_0 + 2\gamma_1 + \cdots + 2\gamma_q] (H-1)^{-1}}}
\]

which needs to be compared to the \( t \)-statistic with \( H - 1 \) degrees of freedom.

### 4.7 Seasonality

Two simple models that account for seasonality when using quarterly data are

\[y_t = a_4 y_{t-4} + \varepsilon_t, \quad |a_4| < 1\]  \hspace{1cm} (4.38)

and

\[y_t = \varepsilon_t + \beta_4 \varepsilon_{t-4} \]  \hspace{1cm} (4.39)

Sometimes these models are easily identifiable from the ACF or the PACF. However, usually identification of seasonality is complicated because the seasonal pattern interacts with the nonseasonal pattern.

Even if the data is seasonally adjusted, a seasonal pattern might remain. Models with multiplicative seasonality allow for the interaction of the ARMA and the
seasonal effects. Consider the following multiplicative formulations

\[
(1 - a_1 L)y_t = (1 + \beta_1 L)(1 + \beta_4 L^4)\varepsilon_t, \quad (4.40)
\]

\[
(1 - a_1 L)(1 - a_4 L^4)y_t = (1 + \beta_1 L)\varepsilon_t, \quad (4.41)
\]

While the first formulation allows the moving-average term at lag 1 to interact with the seasonal moving-average at lag 4, the second formulation allows for the autoregressive term at lag 1 to interact with the seasonal autoregressive effect at lag 4. Let’s rewrite the first one as

\[
y_t = a_1 y_{t-1} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \beta_4 \varepsilon_{t-4} + \beta_1 \beta_4 \varepsilon_{t-5}. \quad (4.42)
\]

We can estimate this type of models in Stata. Consider the following example were we estimate the model in Equation 4.42,

```
use http://www.stata-press.com/data/r11/wpi1
gen dwpi = d.wpi
arima dwpi, arima(1,0,1) sarima(0,0,1,4) noconstant
```

with \( y_t = dwpi \),

```
(setting optimization to BHHH)
Iteration 4: log likelihood = -136.99727
(switching optimization to BFGS)
Iteration 7: log likelihood = -136.99707
ARIMA regression
Sample: 1960q2 - 1990q4 Number of obs = 123
Log likelihood = -136.9971

|                      | Coef.   | Std. Err. | z     | P>|z|   | [95% Conf. Interval] |
|----------------------|---------|-----------|-------|-------|----------------------|
|                      |         | OPG       |       |       |                      |
| ARMA                 |         |           |       |       |                      |
| ar                   |         |           |       |       |                      |
| L1.                  | .9308772| .0382365  | 24.35 | 0.000 | .855935 - 1.005819  |
| ma                   |         |           |       |       |                      |
| L1.                  | -.4521315| .0909366  | -4.97 | 0.000 | -.630364 - .273899  |
| ARMA4                |         |           |       |       |                      |
| ma                   |         |           |       |       |                      |
| L1.                  | .0817591| .0820811  | 1.00  | 0.319 | -.079117 .2426352   |
| /sigma               | .7331657| .0366673  | 20.00 | 0.000 | .6612992 .8050322   |
```

Replacing the estimated parameters in the equation

\[
y_t = 0.931 y_{t-1} + \varepsilon_t - 0.452 \varepsilon_{t-1} + 0.081 \varepsilon_{t-4} - 0.037 \varepsilon_{t-5}. \quad (4.43)
\]

where \(-0.452 \times 0.081 \approx -0.037\), and \( \sigma = 0.733 \).
4.8 ARMAX Models

A simple extension of an ARMA($p, q$) model is to include independent variables in the ARMA structure. This is important because in the traditional ARMA model the dependent variable is only a function of its past values and disturbances. Consider the following example from the Stata manual,

$$\text{consump}_t = \beta_0 + \beta_1 m_2 + \mu_t$$  \hfill (4.44)

where we model the disturbance as

$$\mu_t = \rho \mu_{t-1} + \theta \epsilon_{t-1} + \epsilon_t.$$  \hfill (4.45)

This model can be fitted using

```stata
arima consump m2, ar(1) ma(1)
(setting optimization to BHHH)
Iteration 0: log likelihood = -739.44023
Iteration 41: log likelihood = -717.97025

ARIMA regression
Sample: 1959q1 - 1998q3 Number of obs = 159
Wald chi2(3) = 39847.46
Log likelihood = -717.9703 Prob > chi2 = 0.0000
------------------------------------------------------------------------------
| OPG
| consump | Coef. Std. Err. z P>|z| [95% Conf. Interval]
|-------------+----------------------------------------------------------------
| consump | m2 | 1.002315 .0471436 21.26 0.000 .9099155 1.094715
| _cons | 725.3014 777.8313 0.93 0.351 -799.2199 2249.823
| ARMA | ar | L1. | .9991257 .0054368 183.77 0.000 .9884697 1.009782
| ma | L1. | .420434 .0591463 7.11 0.000 .3045095 .5363586
| /sigma | 21.6242 .915917 .9159176 23.61 0.000 19.82903 23.41936
------------------------------------------------------------------------------
```

4.9 A Model for Canadian Employment

To illustrate the ideas we studied so far, we now look an example using the quarterly, seasonally adjusted index of Canadian employment, 1962.1-1993.4, which is plotted in Figure 4.9. The series appears to have no trend and no remaining seasonality. It does, however, appear to have a high serial correlation. It evolves in a slow and persistent fashion. When computing the Q-statistic (not shown) it is easy to see that is in not white noise.
The sample autocorrelations and the partial autocorrelations are presented in Figure 4.10. The sample autocorrelations are large and display slow one-sided decay. The sample partial autocorrelations at large at first (at displacement 1), but are negligible beyond displacement 2. It is clear the existence of a strong cyclical component. Compare these ACF and PACF with the ones generated with the simulated process $z_t = 0.9 \cdot z_{t-1} + \epsilon_t$ in Figure 4.5.

To select the order of the most appropriate ARMA($p,q$) model, we need to estimate all combinations of the orders $p$ and $q$. A quick way to do this is using the user generated command estout that allows comparing Stata regression outputs side by side:

```stata
ssc install estout
eststo arma00: arima caemp if qtr<121, arima(0,0,0)
eststo arma10: arima caemp if qtr<121, arima(1,0,0)
eststo arma20: arima caemp if qtr<121, arima(2,0,0)
eststo arma30: arima caemp if qtr<121, arima(3,0,0)
eststo arma40: arima caemp if qtr<121, arima(4,0,0)
esttab, aic bic title(Canadian Employment) mtitle("arma00" "arma10" "arma20" "arma31" "arma40")
```

After estimating all the combinations, we finds that based on the BIC, the most appropriate is the ARMA(4,2) model. Notice that we are only estimating the model with data up to $qtr<121$, which corresponds to the first quarter of 1990. After estimating the ARMA(4,2) model we obtain the ACF and the PACF of the residuals to see if they are white noise. These ones are shown in Figure 4.11, which clearly shows that they are white noise. The Q-statistics also corroborates this result.

```stata
arima caemp if qtr<121, arima(4,0,2)
predict resi, res
ac resi, saving(acresi)
```
Autoregressive and Moving-Average Models II

Autocorrelations of \( caemp \)

Partial autocorrelations of \( caemp \)

95% Confidence bands

ACF and PACF for Canadian Employment

Fig. 4.10 ACF and PACF for Canadian Employment

\[
\begin{align*}
\text{ACF and PACF for the ARMA(4,2) residuals} \end{align*}
\]

\[
\begin{array}{l}
\text{pac resi, saving(pacresi)} \\
\text{gr combine acresi.gph pacresi.gph, col(1)} ///
\text{iscale(0.7) fysize(100)} ///
\text{title("ACF and PACF for the ARMA(4,2) residuals" )}
\end{array}
\]

corrgram resi, lags(20)

\[
\begin{array}{l}
\text{LAG} \quad | AC \quad | PAC \quad | Q \quad | \text{Prob>Q [Autocorrelation]} \quad | \text{Partial Autocor} \quad |
\hline
1 \quad 0.0350 \quad 0.0352 \quad 0.1699 \quad 0.6802 \quad | \quad | \\
2 \quad -0.0563 \quad -0.0574 \quad 0.61346 \quad 0.7359 \quad | \quad | \\
3 \quad -0.0294 \quad -0.0266 \quad 0.73508 \quad 0.8649 \quad | \quad | \\
\text{(rest of the output omitted)}
\end{array}
\]

Now, to forecast the values of Canadian Employment we can either use the one-step-ahead forecast of the dynamic forecast. The commands to obtain the forecasted values, the forecast graphs and the graphs of the forecasting errors are:

\[
\begin{align*}
predict \ ehat, \ y \\
predict \ ehatdy, \ dynamic(121) \ y \\
predict \ eer, \ res \\
predict \ eerdy, \ dynamic(121) \ res \\
\text{label variable caemp "Canadian employment"} \\
\text{label variable ehat "One-step-ahead forecast"} \\
\text{label variable ehatdy "Dynamic forecast"} \\
\text{twoway line ehat ehatdy caemp qtr if qtr > 100, ///} \\
\text{title("Canadian Employment Forecasts")}
\end{align*}
\]

\[
\begin{align*}
lable variable eer "One-step-ahead forecast error" \\
lable variable eerdy "Dynamic forecast error" \\
\text{twoway line eer eerdy qtr if qtr > 100, ///} \\
\text{title("Forecast Errors")}
\end{align*}
\]
4.9 A Model for Canadian Employment

The results for the forecasts are shown in Figure 4.12, while the results for the forecasting errors are in Figure 4.13.

Some other statistical packages, like Eviews or Gretl, may be easier to use for some forecasting features. For example, with Gretl we can generate the forecast of Canadian Employment with 95% confidence bands as shown in Figure 4.14.\(^2\)

### 4.9.1 Diebold and Mariano Test

How do we evaluate/rank two different forecasts? A popular test is the one proposed in Mariano and Diebold (1995). This one can be implemented in Stata with the following user generated tool:

```
ssc install dmariano
dmariano caemp ehat e2hat if tin(121,139), max(2) crit(MAE)
```

Diebold-Mariano forecast comparison test for actual : caemp
Competing forecasts: ehat versus e2hat
Criterion: MAE over 19 observations
Maxlag = 2 Kernel : uniform

<table>
<thead>
<tr>
<th>Series</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ehat</td>
<td>1.307</td>
</tr>
</tbody>
</table>

By this criterion, $e_{\hat{t}}$ is the better forecast.

H0: Forecast accuracy is equal.

$S(1) = -3.556$  $p$-value = 0.0004

In this specification we are using Mean Absolute Error (MSE), $g(e_i) = |e_i|$. Notice that if the criteria $\text{crit(MSE)}$ (Mean Squared Error) is selected, we are just using $g(e_i) = e_i^2$. For more details see help dmarioano.

### 4.10 Supporting .do files

For Figures 4.9 and 4.10.

```stata
use employment.dta, clear
tset qtr
label variable qtr "Quarter"
twoway line caemp qtr, ///
title("Canadian Employment Index")
ac caemp, saving(accaemp)
pac caemp, saving(paccaemp)
gr combine accaemp.gph paccaemp.gph, col(1) ///
iscale(0.7) fysize(100) //
title("ACF and PACF for Canadian Employment")
```

The rest of all the ARMA($p,q$) models:
Fig. 4.13 Canadian Employment One-Step-Ahead and Dynamic Forecast Errors

Fig. 4.14 Canadian Employment Forecast using Gretl

```
eststo clear
eststo arma01: arima caemp if qtr<121, arima(0,0,1)
eststo arma11: arima caemp if qtr<121, arima(1,0,1)
eststo arma21: arima caemp if qtr<121, arima(2,0,1)
eststo arma31: arima caemp if qtr<121, arima(3,0,1)
eststo arma41: arima caemp if qtr<121, arima(4,0,1)
```
esttab, aic bic title(Canadian Employment) mtitle("arma01" "arma11" "arma21" "arma31" "arma41")
eststo clear
eststo arma02: arima caemp if qtr<121, arima(0,0,2)
eststo arma12: arima caemp if qtr<121, arima(1,0,2)
eststo arma22: arima caemp if qtr<121, arima(2,0,2)
eststo arma32: arima caemp if qtr<121, arima(3,0,2)
eststo arma42: arima caemp if qtr<121, arima(4,0,2)
esttab, aic bic title(Canadian Employment) mtitle("arma02" "arma12" "arma22" "arma32" "arma42")
eststo clear
eststo arma03: arima caemp if qtr<121, arima(0,0,3)
eststo arma13: arima caemp if qtr<121, arima(1,0,3)
eststo arma23: arima caemp if qtr<121, arima(2,0,3)
eststo arma33: arima caemp if qtr<121, arima(3,0,3)
eststo arma43: arima caemp if qtr<121, arima(4,0,3)
esttab, aic bic title(Canadian Employment) mtitle("arma03" "arma13" "arma23" "arma33" "arma43")
eststo clear
eststo arma04: arima caemp if qtr<121, arima(0,0,4)
eststo arma14: arima caemp if qtr<121, arima(1,0,4)
eststo arma24: arima caemp if qtr<121, arima(2,0,4)
eststo arma34: arima caemp if qtr<121, arima(3,0,4)
eststo arma44: arima caemp if qtr<121, arima(4,0,4)
esttab, aic bic title(Canadian Employment) mtitle("arma04" "arma14" "arma24" "arma34" "arma44")